

CoCoA-5 - Bug #343

Interpreter SEGV

23 Apr 2013 16:43 - John Abbott

Status:	Closed	Start date:	23 Apr 2013
Priority:	High	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Parser/Interpreter	Estimated time:	0.00 hour
Target version:	CoCoA-5.0.3	Spent time:	4.05 hours
<b>Description</b>			
I get a SEGV when I send the following file to CoCoA-5 from an Emacs buffer (using send-file):			
<pre>For j := 1 To 200 Do   For k := 1 To 200 Do     EndFor;</pre>			
Presumably the interpreter gets confused by the premature EOF.			

History

#1 - 24 Apr 2013 08:09 - Anna Maria Bigatti

I get the same error calling directly

```
./CoCoAInterpreter < tests/bug-emacs.cocoa5
```

With this line only

```
For j := 1 To 200
```

I get

```
Assertion failed: (px != 0), function operator->, file /Users/bigatti/0.99/boost/boost/smart_ptr/intrusive_ptr
.hpp, line 166.
ERROR: I was expecting "Do" but I've found "<end of file>"
Process cocoa5 abort trap
```

## #2 - 24 Apr 2013 08:19 - Anna Maria Bigatti

In any case I get abort trap, not segv

## #3 - 27 Apr 2013 22:40 - John Abbott

The problem can be reproduced simply by reading in a file containing just the character **1**. It does not matter whether there is a newline or not.

Evidently the interpreter does not handle premature EOF properly.

**Note** it does work OK if the input file contains `//1` or `--1`

## #4 - 27 Apr 2013 23:06 - John Abbott

Well, it's not all Apple's fault... there was a "surprise" in a Makefile (sigh).

## #5 - 28 Apr 2013 10:58 - John Abbott

- Status changed from New to In Progress
- Assignee set to John Abbott
- % Done changed from 0 to 50

I have found a *workaround*.

The problem surfaces in `reportLineNumberWhenMeaningful` which tries to follow a NULL pointer when `fromPos` is set to `CharPointer::Null` -- this value was set inside `Lexer::getCP` at line 85 in `Lexer.C`

The workaround is simply to add the following as a first line in the defn of `reportLineNumberWhenMeaningful`

```
if (fromPos==CharPointer::Null) return false;
```

This avoids the SEGV but is not a proper solution; to find a proper solution I'd like a little help from Giovanni -- Giovanni? [preferably fairly urgently, Giovanni?]

## #6 - 29 Apr 2013 09:00 - John Abbott

A suggestion for modifying the error message produced when EOF is found in the middle of a command.

Currently the error messages produced look like this:

```
ERROR: Expecting a semicolon, to end the statement, or an assignment operator
<End of file>
```

```
ERROR: Unexpected End-Of-File while reading "/Users/abbott/bug.cocoa5"
```

It is not clear to me what is the point of printing <End of file> on a line by itself. To me it looks like unhelpful clutter.

I think we could produce something more readable/comprehensible. I suggest something like the following:

```
ERROR: Incomplete command (reached end of input)
ERROR: Expecting a semicolon,...
```

**Note** The strange <End of file> message might be produced by `DefaultErrorReporter::outputUnderlinedChars` in `Lexer.C:641`. There is something similar in function `IdeErrorReporter::outputHighlightedChars` in file `C5.C:648`

#### #7 - 06 May 2013 14:42 - John Abbott

- Status changed from *In Progress* to *Feedback*
- % Done changed from 50 to 80

After speaking to Anna we decided to make "minimal" changes.

Instead of referring to *end of file* messages refer to **end of input** (we think this will be more easily understood when input is from `SourceRegion`).

I have suppressed the printing of "context" (in `DefaultErrorReporter::outputUnderlinedChars`) if the context is EOF; I found the old behaviour unhelpful and confusing -- indeed I even thought there was a bug which spuriously caused <End Of File> to be printed.

#### #8 - 24 May 2013 14:41 - John Abbott

- Status changed from *Feedback* to *Closed*
- % Done changed from 80 to 100

I have accepted the "hack" added to `reportLineNumberWhenMeaningful` (see source line `lexer.C:679`).

While testing I also improved robustness of the `SourceRegion` command, and modified the error messages so that they are more comprehensible.

Since no further problems have arisen in the last 18 days, I'm closing this issue.